AMENDMENTS TO THE SPECIFICATION

Please amend the specification, as follows:

Replace paragraph [0017] with the following amended paragraph [0017]:

~~$R = A*B*r^{-1} \text{mod } M$, (where the radix $r = 2^n$)~~

$S_N = A*B*R^{-1} \text{ mod } M$, (where the radix $R = 2^n$) (5)

Replace paragraph [0024] with the following amended paragraph [0024]:

where $b_I A (= PP_I)$ is a partial product; $q_I M (= MM_I)$ is a multiple of modulus M which makes one least significant bit (LSB) of SPP_I ($= S_I + PP_I$) into a zero(0) value; [[N]] n is the bit length of modulus M; $N = n/2$; S_I is the partial accumulated result of a previous cycle; S_{I+1} is the partial accumulated result of the current cycle with [[N]] n bits; and S_N is the final computation result. An exemplary radix-4 Montgomery iterative modular multiplication algorithm is:

Replace paragraph [0030] with the following amended paragraph [0030]:

where $N = n/2$; $(-M*M') \text{ mod } 4 = 1$; and –M is the 2's-complement of M. Both the radix-2 and the radix-4 process are iterative processes producing iterative data; data whose value changes with iterations within the loop of $I = 0$; $I < N$; I++. The modular operation speed affects the system performance. Therefore, if the bit length is very long, the system performance is

degraded. To compute MM_I (= q_IM), first the PP_I(= b_IA) is computed and then the computed PP_I and S_I are added. Therefore, [[Power]] power consumption is increased because the accumulator executes the logical computation twice.

Replace paragraph [0039] with the following amended paragraph [0039]:

In exemplary embodiments of the present invention, an average ~~hamming~~ Hamming distance is reduced, where the ~~hamming~~ Hamming distance is the number of different values of the same bit position. Thus, fewer bit changes can result in the reduction of the variation of fan-out (multiplexer) loading.

Replace paragraph [0051] with the following amended paragraph [0051]:

The Modulus processor 300 can output multiple signals (e.g., a multiple modulus selection signal SEL_MM[1:0], a multiple modulus enabling signal EN_MM, a multiple modulus negation signal NEG_MM, . . .). These signals may be stored in register 230. For example, multiple modulus selection signal SEL_MM[1:0] may be stored in sub-register 62, while multiple modulus enabling signal EN_MM may be stored in sub-register 63. In an exemplary embodiment of the present invention, the Modulus processor 300 and multiplexer 10 are used to select multiple modulus values (MM_I) values (e.g., 2M, M, 0, -M, . . .) to supply to the accumulator 100. To select MM_I values, the Modulus processor 300 outputs multiple modulus selection signal SEL_MM[1:0] to the multiplexer 10[[,]]. The multiplexer 10 receives the value of the Modulus M and ~~which uses the value of~~ SEL_MM[1:0] ~~to select~~ and outputs a

value ~~of MM_I (e.g., 2M, M, 0, -M, . . .). The multiplexer (MUX) 10 inputs the modulus M and the two LSBs of the multiple modulus selection signal SEL_MM[1:0] and~~ to AND gate 31. The AND gate 31 receives the input from the multiplexer 10 and a multiple modulus enabling signal EN_MM from the Modulus processor 300. The AND gate 31 then outputs the value of [[MMI]] MM_I. The multiple modulus negation signal NEG_MM and MM_I are combined at the accumulator 100, where NEG_MM is used to indicate bit-inversion, obtaining a MM_I value of -M.

Replace paragraph [0052] with the following amended paragraph [0052]:

Each MUX operation consumes power and energy, hence when a new SEL_MM[1:0] value is used, a MUX operation is performed to change the settings and select a [[MMI]] MM_I value. Using the previous value of SEL_MM[1:0] results in no change in settings and thus no MUX operation. Reducing the necessary number of MUX operations in MM_I's selection decreases the overall power consumption of the multiplier 1000.

Replace paragraph [0053] with the following amended paragraph [0053]:

In exemplary embodiments of the present invention, the Modulus processor 300 further includes a multiple modulus feedback register 61 and a Modulus recoder 70. The feedback register 61 stores the value of SEL_MM[1:0] of the previous iteration as the value SEL_MM_D[1:0]. When the value of [[MMI]] $MM_I = 0$ is desired, the modulus processor 300 outputs a multiple modulus enabling signal, EN_MM, with a value of 0. The signal EN_MM is

input to an AND gate 31. The AND gate 31 inputs the output of the multiplexer 10, which uses the previous value of the multiple modulus selection signal SEL_MM_D[1:0], hence using no MUX operations, and the AND gate 31 outputs a value of $MM_I = 0$. The assignment of $MM_I = 0$ without a MUX operation decreases the power consumption of the multiplier 1000. A coding scheme similar to that described above is shown in Figure 3.

Replace paragraph [0056] with the following amended paragraph [0056]:

To select PP_I values, the Booth processor 301 outputs a partial product selection signal SEL_PP[1:0] to the multiplexer 20. The multiplexer 20 receives the value of the multiplicand A [[(A)]] and partial product selection signal SEL_PP[1:0] and outputs a value to an AND gate 32. The AND gate 32 receives the input from the multiplexer 20 and a partial product enabling signal EN_PP from the Booth processor 301. The AND gate 32 then outputs the selected value of the partial product (PP_I), which is supplied to the accumulator 100. Analogous to the procedure as discussed above, the Booth processor 301 may include a Booth recoder 80 and a partial product feedback register 64. A zero value of PP_I can be selected by storing SEL_PP_D[1:0], a previous value of SEL_PP[1:0], in the partial product feedback register 64. When the value of PP_I=0 is desired, the Booth processor 301 outputs a partial product enabling signal, EN_PP, with a value of 0. The signal EN_PP is input to an AND gate 32. The AND gate 32 inputs the output of the multiplexer 20, which uses the previous value of the multiple modulus selection signal SEL_PP_D[1:0], hence using no MUX operations, and the AND gate 32 outputs a value of [[PPI]] $PP_I = 0$. The assignment of $PP_I = 0$ without a MUX operation

decreases the power consumption of the multiplier 1000. A coding scheme similar to that described above is shown in Figure 4.

Replace paragraph [0057] with the following amended paragraph [0057]:

Figure 4 illustrates a coding scheme in accordance with exemplary embodiments of the present invention. Although Figure 4 shows three inputs to the Booth ~~recoder~~ Recoder 70, the present invention can have a variety of inputs and outputs depending upon the design criteria[[, for]]. For example, Figure 2 shows additional inputs SEL_PP_D[1:0] and A[1:0]. The coding scheme in Figure 4 illustrates that for values of EN_PP = 0, the selected PP_I value is 0, and the value of SEL_PP[1:0] = SEL_PP_D[1:0]. Output values of PP_I[0] shown include 0 and A[0], while output values of PP_I[1] shown include 0, A[0], and A[1]^A[0] (A[1] exclusive-OR A[0]).

Replace paragraph [0058] with the following amended paragraph [0058]:

Additionally, in exemplary embodiments of the present invention, a coding scheme, an example of which is illustrated in Figure 4, reduces an average ~~hamming~~ Hamming distance. The ~~hamming~~ Hamming distance is the number of different values of the same bit position. For example, if SEL_PP[1:0] has the value "00", corresponding to a PP_I value of "A", in the (I-1)-th iteration then a value of SEL_PP[1:0] in the I-th iteration of "11", corresponding to a PP_I value of "2A", results in a ~~hamming~~ Hamming distance of two (2). It is desirable to reduce the ~~hamming~~ Hamming distance between two values and thus the number of bit inversions and computation power usage. By selecting a coding scheme where if a previous iteration PP_I value

is A or 2A, then the subsequent iteration PP_I value is restricted and can not be 2A and if a previous iteration PP_I value is -A or -2A, then the subsequent iteration PP_I value is restricted and can not be -2A. Figure 4 illustrates various bit values for the coding scheme, however the coding scheme of exemplary embodiments of the present invention should not be interpreted to be limited to the bit pattern shown in Figure 4. For example, a value of PP_I for "A" can correspond to a SEL_PP[1:0] value of 10 instead of the shown 00.

Replace paragraph [0059] with the following amended paragraph [0059]:

A similar ~~hamming~~ Hamming distance coding scheme, as used for the selection of [[PPI]] PP_I values discussed above, can be applied for the selection of values of [[MMI]] MM_I.

Replace paragraph [0064] with the following amended paragraph [0064]:

In an exemplary embodiment of the present invention, a multiple modulus synchronization register 240 and/or a partial product synchronization register 220 are provided. Syncronization registers 220 and 240 use reverse clock phase with respect to clock phase of other registers in multiplier 1000. The multiple modulus synchronization register 240 may store values of SEL_MM[1:0] and EN_MM in sub-registers 62 and 63 respectively. If a partial product synchronization register 220 is used, it may store values of SEL_PP[1:0] and EN_PP in sub-registers 68 and 69 respectively. One or both synchronization registers can be used and the discussion herein should not be interpreted to limit the exemplary embodiments of the present invention to one synchronization register. In exemplary embodiments where both

synchronization registers are used, SEL_PP[1:0] and SEL_MM[1:0] are stored in sub-registers 68 and 62, respectively, while EN_PP~~[1:0]~~ and EN_MM~~[1:0]~~ are stored in sub-registers 69 and 63, respectively. In response to a clock signal CK, SEL_PP[1:0] is input to multiplexer 20 substantially at the same time as [[and]] SEL_MM[1:0] ~~are substantially simultaneously~~ is input to ~~multiplexer 20 and~~ multiplexer 10[[,]]. ~~respectively, while~~ Similarly, in response to the clock signal CK, EN_PP is input to AND gate 32 substantially at the same time as [[and]] EN_MM ~~are substantially simultaneously~~ is input to AND gate 31 ~~gates 32 and 31, respectively~~. The outputs of the multiplexers 20 and 10 are generated substantially at the same time. Similarly, [[and]] the outputs of AND gates 32 and 31 ~~and 32~~[[,]] are generated ~~simultaneously~~ substantially at the same time. Thus, MM_I and PP_I are synchronized and supplied to the accumulator 100. Thus one logical operation can be performed per data set MM_I and PP_I, as opposed to the conventional two logical operations, significantly decreasing the power consumption of multiplier 1000.

Replace paragraph [0067] with the following amended paragraph [0067]:

The description of the invention is merely exemplary in nature and, thus, variations that do not depart from the gist of the invention are intended to be within the scope of the embodiments of the present invention. Such variations are not to be regarded as a departure from the spirit and scope of the present invention. For example multiplexers 10 and 20 can have a variety of ratio values. The multiple modulus synchronization sub-register 62 can function a dual purpose as a multiple modulus feedback register without having an additional separate multiple modulus feedback register 61 [[230]]. Likewise, partial product feedback register 64

can serve a dual purpose as a sub-register of the partial product synchronization register 220, thus removing the need for both a sub-register 68 and a feedback register 64. In other variations the synchronization registers 220 and 240 can be used without other registers such as a pipeline register and/or feedback registers.